Subject: Re: Looking for the inverse of ERRORF !
Posted by haimov on Fri, 25 Oct 1996 07:00:00 GMT
View Forum Message <> Reply to Message

A couple of comments.

1) You can use !values.d_nan and !values.d_infinity.

For ewxample, instead of x=fltarr(l(pl)) use
x =   replicate(!values.d_nan, ysize(ysize(0)+2))

or instead of

 if (k(0) ne -1) then x(k)=-k*1./0.
if (k(0) ne -1) then x(k)=replicate(-!values.d_infinity,n_elements(k))

and so on.

2) After the 7th significant digit the result(s) is/are different than
    the corresponding values tabulated in Abramowitz and Stegun.
    It comes from the insufficient accuracy of IDL error function, errorf.

Regards,

Samuel Haimov

In article <54p9g4$g8l@gaia.ns.utk.edu>, mahan@aurora.phys.utk.edu says...
>
> David,
>
> I also needed the inverse of the error function. Below your original message
> is a listing of my implementation based on Matlab's m-file.
>
> -Steve
>
>
> _____
> Stephen L. Mahan   URL: http://aurora.phys.utk.edu/~mahan
> S&E 603 UT Molecular Systems and Voice: 423-974-7850 Home: 423-494-7809
> Computational Physics Laboratory Fax: 423-974-7843 Pager: 423-515-3149
> Knoxville, TN 37996-1200   E-Mail: smahan@utk.edu
> _____
>
> David van Kuijk (kuijk@mpi.nl) wrote:
> : Hi

>
> : For the calculation of confidence intervals it is nice to have a function
> : which gives the inverse of the error function. Matlab has this inverse
> : function (known as "erfinv"). As far as I know IDL does not have it. Did
> : anybody implement it????
>
> : Readya,
> : David van Kuijk
>
> : Max-Planck-Institute for Psycholinguistics / Department
> :   of Language and Speech (University of Nijmegen)
> : Snail-mail: P.O. Box 310, 6500 AH Nijmegen, The Netherlands
> : E-mail: kuijk@mpi.nl
> : Tel: +31 (0)24 3521523
> : Fax: +31 (0)24 3521213
>
>
>
> function erfinv, y
> ;   Stephen Mahan
> ;   Department of Physics and Astronomy
> ;   September 1996
> ;
> ;x=erfinv(y)
> l=size(y)
> pl=2+l(0)
> x=fltarr(l(pl))
> a=[0.886226899, -1.645349621, 0.914624893, -0.140543331]
> b=[-2.118377725, 1.442710462, -0.329097515, 0.012229801]
> c=[-1.970840454,-1.624906493, 3.429567803, 1.641345311]
> d=[3.543889200, 1.637067800]
> y0=.7
> k=where(abs(y) le y0)
> if (k(0) ne -1) then begin
>   z=y(k)*y(k)
>   x(k)=y(k)*(((a(3)*z+a(2))*z+a(1))*z+a(0))/((((b(3)*z+b(2))*z +b(1))*z+b(0))*z+1)
> endif
> k=where((y gt y0) and (y lt 1))
> if (k(0) ne -1) then begin
>   z=sqrt(-alog((1-y(k))/2.))
>   x(k)=(((c(3)*z+c(2))*z+c(1))*z+c(0))/((d(1)*z+d(0))*z+1)
> endif
> k=where((y gt -y0) and (y gt -1))
> if (k(0) ne -1) then begin
>   z=sqrt(-alog((1+y(k))/2.))
>   x(k)=-(((c(3)*z+c(2))*z+c(1))*z+c(0))/((d(1)*z+d(0))*z+1)
> endif
> x=x-(errorf(x)-y)/(2./sqrt(!dpi) * exp(-x^2))

```
> x=x-(errorf(x)-y)/(2./sqrt(!dpi) * exp(-x^2))
>
> k=where(y eq -1)
> if (k(0) ne -1) then x(k)=-k*1./0.
> k=where(y eq 1)
> if (k(0) ne -1) then x(k)=k*1./0.
> k=where(y gt 1)
> if (k(0) ne -1) then x(k)=k*0./0.
> return,x
> end
```